Interoperable 3D Emergency Routing Based on OpenLS

Pascal Neis, Arne Schilling, Alexander Zipf
i3mainz, Institute for Spatial Information and Surveying Technology
University of Applied Sciences FH Mainz

Summary

In this paper we introduce the concept and the implementation of an Emergency Route Service 3D (ERS3D), which is based on the already established OGC Standards OpenLS and which further covers the following two aspects: Firstly, the possibility to bypass restricted areas or closed streets during disaster situations and secondly the ability to return a 3D route geometry that can be integrated into a 3D application. Within the project OK-GIS (Open and free GIS for Disaster Management) a Route Service based on OpenLS Specifications was implemented. This was then taken as the basis for building more complex services. Based on the initial Route Service an Emergency Route Service (ERS) was developed, along with a 3D Route Service (3DRS). The combination of the 3DRS and the ERS then resulted in a 3D Emergency Route Service (3DERS). An important aspect is that this can be achieved solely using OGC standards, so no further standards need to be specified for this application. Moreover this shows that more complex services can be realized by combining existing OGC services and integrating them into a 3D spatial data infrastructure.

1 Introduction

Unfortunately disasters do not stop at administrative- or system boundaries, therefor special focus needs to be laid on the development of an open spatial data infrastructure (SDI) for disaster management. This should improve the opportunity to use the systems, offered by different institutions or regions, in the long term in a more interoperable way. For geographic information services, open interfaces should be deployed such as the ones defined by OGC or ISO (Weiser et al. 2006).

The current activities with respect to spatial data infrastructures (SDI) underpin the utilization of open standards for enabling the communication between components offered by different organizations or service providers.. Here, the Open Geospatial Consortium (OGC) plays a primary role by offering a substantial range of standards for GI-related services. Open source SDI-frameworks such as deegree (Fitzke et al. 2004) already offer a number of components which can be used to build up spatial data infrastructures. For disaster management scenarios, the OGC OpenLS Service, amongst others, is of particular importance for providing route services.. In this paper focus is put on one specific scenario:

Rescue workers or citizens need to by-pass non accessible or restricted areas, for example due to flooding, a bomb threat or other exposures. This information is essential for relief personnel as well as for the affected population. A corresponding service has been implemented in the OK-GIS project (http://www.okgis.de) and integrated into the specific SDI for disaster management. Its relevance has been verified by a scenario- and request analyses conducted by Biermann et al. 2005 within the project.

Web services for routing in disaster management situations have already been tested as prototypes by Keßler et al. (2003): amongst others also a proprietary *Emergency Routing Service* was realized - however without using OGC standards at that time.

Further requirements for routing have been added by the recently enforced standardization of 3D visualizing services and 3D city models. Especially the concepts of visualization and user guidance of the routing can not be applied directly to 3D systems. Depending on the application many different versions can be ideal (e.g. navigation of vehicles, pedestrians, web applications). The routing service can however support the 3D application by calculating the correct position of the route geometry.

In addition to the familiar on-board car navigation systems provided by commercial manufacturers, there is a range of commercial applications for PDA´s available, which support map based route calculation and navigation assistance. Most of these are also primarily for vehicle navigation. Many of the offered devices already feature a so-called “3D” mode, which visualizes the maps in panorama perspective (pseudo 3D), but without real 3D data like DEMs or 3D buildings. As of now, the use of real 3d models for routing has been realized at most as prototype applications and evaluated for smaller urban areas.

2 The OpenLS Route Service

Since the year 2000, the *Open Location Services* (OpenLS) initiative of the OGC has been developing free specifications (interfaces and protocols), for standardizing services relevant for *Location Based Services* (LBS). The *OpenLS Service Framework* consists of five core services. Further details can be found in Zipf (2004) und Neis (2006, 2007). A 6th service (Tracking Service) is just being added to a new version of the specification.

- The *Directory Service* is a service that provides access to an online directory (e.g., Yellow Pages) to find the location of a specific or nearest place, product or service.
- The *Gateway Service* is a service that fetches the position of a known mobile terminal from the network; this interface is modeled after the Mobile Location Protocol (MLP), Standard Location Immediate Service.
- The *Location Utility Service* provides a Geocoder/Reverse Geocoder; the Geocoder transforms a description of a location, such as a place name, street address or postal code, into a normalized description of the location with a Point geometry usually placed using Cartesian coordinates, often latitude and longitude.
- The *Presentation (Map Portrayal) Service* portrays a map made up of a base map derived from any geospatial data and a set of Abstract Data Types as overlays.
- The *Route Service* determines travel routes and navigation information according to diverse criteria.

These services are being implemented in the OK-GIS project in Java. The first realization is the Route Service. The OpenLS Core Service 5 - the Route Service (RS) - allows route planning on a street network. The OpenLS specification defines how the RS is to be addressed, which options it will provide and in what form the answers are to be sent back. Besides start- and destination points various other criteria can be entered (e.g. time, distance, travel type, etc.). Even start- and destination points can be set in multiple different ways (e.g. through differnet types of address information, coordinates, points of interest, geometries, etc.). The alternatives for the output are even more extensive:

- *1. RouteSummary* - general information, such as total route and total route time.
- *2. RouteGeometry* - Geometry of the calculated route (line with all way points)

- *3. RouteInstruction* - "Step by step" driving- or walking instructions of the calculated route. This was realized in our application in a simple form for various languages.
- *4. RouteMaps* - Maps, in which the calculated route is displayed. Amongst other possibilities it is possible to add several *RouteMaps* at the same time. for example also an overview map as well as detailed maps of the start- and destination points can be returned.

The implemented route service (RS) (Neis 2006) supports all mandatory and most optional parameters of the OpenLS RS specification. One way street information is also supported by this service. The algorithm used, is the well known Dijkssra Algorithm (Dijkstra 1959). Further the Levenshtein Algorithm (Ringlstetter 2003) from comuter linguistics has been applied in order to avoid spelling or typing mistakes of addresses or POI´s. One advantage of the OpenLS route service specification, is the possibility to include an *AvoidList* into the route request. In this, a list of areas or streets can be implemented which should be by-passed or avoided. The RS will identify the optimal route between start and destination point, incorporating the areas to be avoided and giving by-pass information for this area. The *Emergency Route Service* (ERS) uses this function in order to avoid danger zones and blocked or closed roads in case of a disaster situation.

3 Interoperable Route Planning in Disaster Management – an OpenLS conform Emergency Route Service (ERS)

3.1 The Problem

Emergency personnel such as the fire brigade have to be able to access the areas affected by an disaster as soon as possible. The time it takes to get from the station to the affected area is essential for rescue success. All possible hindrances such as traffic jams, heavy traffic or blocked roads/areas cost valuable time. For this reason, all obstacles in the way of the relief force should be incorporated, into the routing so that these may be considered and by-passed.

Furthermore, the ERS should not only give support in emergency situations, but also in disaster prevention. For example it may be possible to estimate the effect a possible bomb threat might have on the traffic, before such an event occurs. By evaluating such simulations, the police is then able to effectively plan a by-pass area.

3.2 Automated By-Passing of Danger Zones through ERS

An ERS is a more specific RS which *automatically* includes up to date information about current danger zones and closed roads in the route calculation and gives information on how to by-pass *avoid areas*. Queries towards the ERS are carried out the same way as queries towards the OpenLS RS. The ERS adds dangerous area- and closed road information from a central database (resp. in the frame of a SDI for disaster management, a WFS), mainainted by a central control unit, to the original request and to a new or an existing *AvoidList*. The altered request is then transferred to the normal RS. The RS calculates the route considering the *AvoidAreas* and returns the OpenLS specific *RouteInstructions*, *RouteGeometry* and *RouteMaps*.

Fig. **1** shows the sequence diagram of the ERS. The components we developed (ERS, OpenLS RS) have been combined with already available open source package components such as *Web Feature Service* (WFS) and *Web Map Service* (WMS) and then accessed via an internet protocol (HTTP post). Present danger zones and closed roads can be handled by the ERS in different ways. In our case, these are being maintained in a WFS. In the actual deployed system, this would be done by the control unit. The bounding box of the *GetFeatures* request is defined by the spatial extent of the route start and end point. Every start- or destination address has to be geocoded before hand. In our case this is done internally, via an internal database. Alternatively an *OpenLS Location Utility Service (Geocoder/Reverse Geocoder)* could also be used instead in order to solve this task also in a standards-based manner.

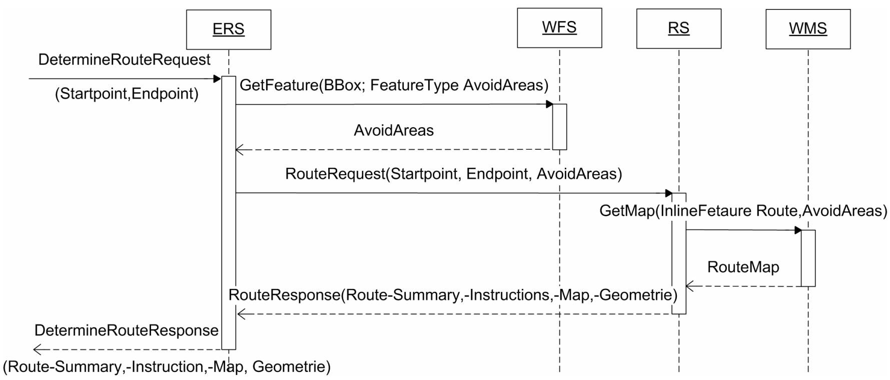

Fig. 1: Emergency Route Service Sequence diagram.

If a road is blocked or a whole area can or should not be accessed, a geometric intersection is calculated between the polygon of the danger zone and the street network in order to detect which streets are not accessible.

All mentioned services (RS, ERS, 3DRS, 3DERS) use the same standardized interface, the Open LS Route “RouteRequest”. An example is given in Listing 1.

```
<xls:XLS ... >
  <xls:RequestHeader/>
  <xls:Request methodName="RouteRequest" requestID="1" version="1.1">
    <xls:DetermineRouteRequest>
      <xls:RoutePlan>
        <xls:RoutePreference>Fastest</xls:RoutePreference>
        <xls:WayPointList> ... </xls:WayPointList>
        <xls:AvoidList>
          <xls:AOI>
            <gml:Polygon>
              <gml:exterior>
                <gml:LinearRing xsi:type="gml:LinearRingType">
                  <gml:pos>3434774.787 5794688.517</gml:pos>
                  <gml:pos>3434932.208 5794547.765</gml:pos>
                  <gml:pos>3434678.483 5794249.592</gml:pos>
                  <gml:pos>3434506.247 5794384.788</gml:pos>
                  <gml:pos>3434774.787 5794688.517</gml:pos>
                </gml:LinearRing>
              </gml:exterior>
            </gml:Polygon>
          </xls:AOI>
          <xls:Address xsi:type="xls:AddressType" countryCode="DE-NI">
            <xls:StreetAddress>
              <xls:Building xsi:type="xls:BuildingLocatorType" number="50"/>
              <xls:Street officialName="Am Stollenbach"/>
            </xls:StreetAddress>
            <xls:Place type="Municipality">Osnabrück</xls:Place>
            <xls:PostalCode>49074</xls:PostalCode>
          </xls:Address>
        </xls:AvoidList>
      </xls:RoutePlan>
    </xls:DetermineRouteRequest>
  </xls:Request>
</xls:XLS>
```

Listing 1: Example of an OpenLS RS RouteRequest

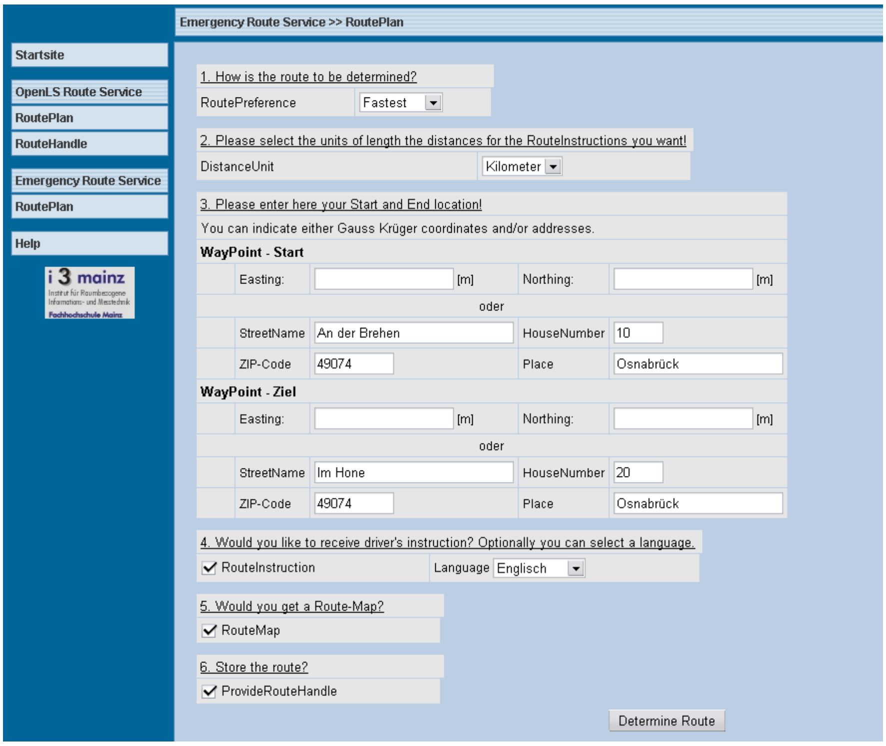

Fig. 2: Prototype Test User Interface of the ERS.

4 3D Route Planning based on a 3D-SDI

4.1 3D-SDI - the example of Heidelberg

The introduced route services are currently being integrated into a larger framework and are being evaluated regarding their usability for particular applications. The main focus lies on interoperable use and visualization of 3D city models building a 3D spatial data infrastructure based on OGC services (http://www.3d-gdi.de). Among the suggestions recently discussed by the OGC, the 3D web service (W3DS) plays an important role so that in the near future it seems realistic to be able to process 3D data via OGC services (Schilling et al. 2007). The W3DS is similar to the already standardized Web Terrain Service (WTS)(Web Perspective View Service, WPVS). The difference is that it doesn't only deliver rendered 3D images but 3D geometries resp. scenes in standardized exchange formats which can be viewed by using any internet browser and corresponding plugin. Similar to the WMS, the bounding box and the name of the layers styles which are recognized by the service are the input parameters, which are sent to the W3DS. Further parameters include parameters for specifing the viwing direction etc, such as the Point of Interest (POI), Point of Camera (POC), Angle of view (Pitch, Yaw, Roll). The combination between W3DS with other OGC services like catalog services, WFS, WMS, web processing services and others, enable the realization of a geodata infrastructure for 3D geodata (3D-GDI). This is also one of the goals of our Heidelberg City project (www.heidelberg-3d.de). Using a custom Java3D-client, which can be started via Java Webstart, various OGC services can be activated and the outcomes can be displayed in 3D. Through the integration of the route service, planning and visualizing virtual tours within the 3D city model is now possible (similar to Schilling and Zipf 2002). At the same time, further clients based on MS.NET platform (Bock 2007, in progress) and on mobile devices (Fischer et al 2006) have been developed which access the same infrastructure.

Beside the frequently stated application scenarios such as tourist information systems, urban planning and -marketing, support should also be offered for disaster management. A first step can be seen in the integration of the emergency route service and the adequate presentation of the routes in 3D. For this, the Java3D-Viewer has been enhanced so that it can access the ERS3D. For realizing the ERS3D itself, the route geometry calculated by the RS has to be adjusted to the DEM already available. This process is further explained in the next paragraph. Other geodata such as buildings, landmarks, forest areas, grasslands, building blocks, water, streets, trees, traffic lights and traffic signs are requested by the W3DS as VRML models, which can then be visualized directly in the Java3D Viewer.

4.2 RS3D – an OpenLS 3D Route Planner

Route services are based on topology graphs. The fact that a certain geometry can be assigned to each route segment is only relevant for visualization purposes.. Normally the coordinates used are 2 dimensional (2D). but we, need 3D visualization. This means the route has to be superimposed onto a DEM. This can be done in two ways: One way is that the client uses the 2D geometry and adjusts it to the DEM during visualization. This is only possible for fat clients however, because this function is e.g. not supported by standard software likeVRML browsers. Although an independent client (Java-WebStart-Application) is used, which could be

„taught“ this functionality, it would mean to incorporate relatively heavy computations into the client, which is not desirable. Especially for mobile 3D clients on Smart phones or PDAs (cp. Fischer et al 2006) this is too complex and should already be pre-processed by the server.

In principle an RS can also deliver 3D coordinates. The returned GML coordinates simply have 3 instead of 2 coordinate values. In order to realize this, information about the terrain near the street network is needed. One boundary condition is that not every point on the route geometry can be assigned permanent elevation information. That is why a dynamic alignment to the DEM is needed. The realson for this is that the DEM is also dynamic, as different resolutions have to be supported according to the level of detail (LOD) which is dependent from the application, network and client resources, etc. (cp. Schilling and Zipf 2003).

Fig. 3: The Route in the 3DViewer. Within a seperate window the typical Open LS information like RouteSummary, RouteInstructions and RouteMap are displayed (Data: Land Surveying Office Heidelberg and EML Heidelberg).

Also, the points of the route geometry, calculated by the RS, are not necessarily static but can be changed by generalization methods or extensions, such as user generated specified generalization (Kray et al. 2003), in order to generate with respect to the language ideal driving instructions (*Route Instructions*). Further recent work on the topic of cognitive adequate support for navigation is presented by Hansen et al. 2006, who suggest extensions to the OpenLS standard in order to incorporate landmarks in the route descriptions. But the goal of our RS3D implementation was to stick to the OpenLS standard as it is available right now, without having to change the standard itself. This way the RS3D can be integrated into the 3D-SDI of Heidelberg without major efforts. In fact, the interface of the RS3D does not differ from the specification of the OpenLS RS. The process of adjusting the route geometry to follow the elevation as defined by the DEM is done internally. The result is that the RS3D returns coordinates of the GML-LineStrings that contain additional height values.

Since a WFS is used as a data source for the DEM, our goal of staying OGC conform as much as possible could be entirely fulfilled. A possible future extension could be to outsource the interpolation function to a *Web Processing Service* (WPS) (cmp. Kiehle et al. 2006), where the points are then calculated to the DEM. This would make the application more modular - but at the cost of performance.

The implementation of the RS3D was carried out as a cascade of OpenLS RS as well as of further OGC services. Orchestrating such service chains through standards like BPEL is a current issue (Weiser and Zipf 2007). Requests to the RS3D are carried out the same way as to the OpenLS RS. The requests are forwarded to a normal OpenLS RS which carries out the actual calculations. The DEM, which is in the same area (OGC-Filter function *DWITHIN*) as the geometry of the route segments, is accessed by a WFS via *GetFeature* (using a PostGIS-DB).

Although it would be possible to simply identify a height value on the DEM for every coordinate, this would result in visible errors like intersections with bumps quite often, For this reason the exact course of the DEM is calculated for all route segments. The DEM is a TIN consisting of the topological elements vertices, edges, and triangles. A spatial index (Quadtree) enables fast access for spatial queries (touches, intersects, contains, etc.). This index is used to detect the topological edges of the tin, which lie directly under the route segments. Via the coordinates of the vertices, the heights of the 2D intersection points of the segment are interpolated with the edge of the DEM. The new points are inserted, in the right order, into the newly created GML *Linestring,* which describes the route geometry. The amount of data needed for the geometric description of the route is dependent on the resolution of the DEM. The processed GML *Linestring* is included in the reply of the OpenLS RS and is returned as part of the result of the RS3D. Then the created 3D route can be displayed along with the rest of the city model by the client - in our case as a pipe-like geometry using a "sweeping" command, which extrudes a profile along the path **(**Fig. 3). Further output elements such as *RouteInstructions, RouteSummary* and

RouteMaps are displayed in a separate window. On top of this, an animation sequence resp. camera tour, is run automatically along the route. This showed however, that the route segments are too fine for animation, since a keyframe is assigned to every segment. As soon as there are several smaller segments, abrupt changes in direction take place, between the keyframes. In order to avoid this, the route geometry is generalized. For this, the Douglas Peucker Algorithm has been applied, which also incorporates z values. This way, many of the smaller and irrelevant segments are filtered out, providing better visualization. The sequence diagram to the RS3D is similar to the following ERS3D (Fig. 4).

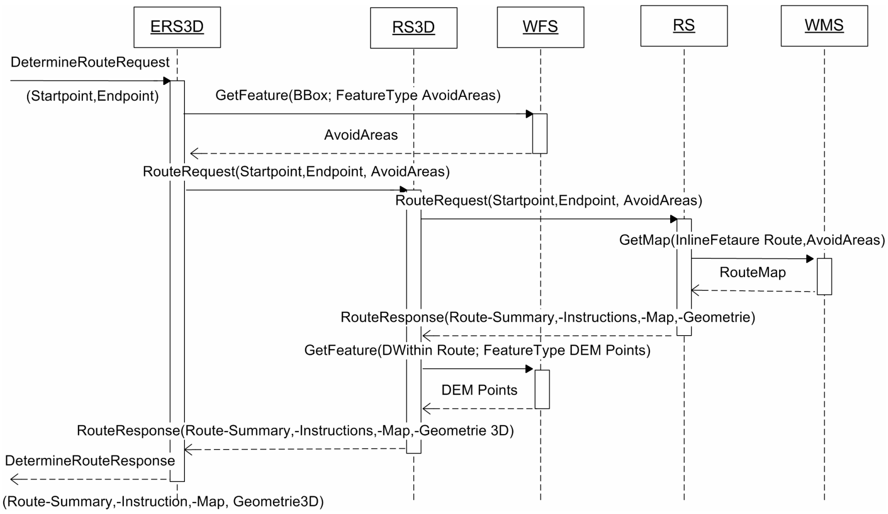

Fig. 4: Emergency Route Service 3D sequence diagram (Detail).

5 Combining the RS3D & ERS: The ERS3D

The RS3D and ERS routing services described above and the concepts they implement can be used in a parallel way, because they can be installed on separate servers. Or they can be eventually aggregated to an Emergency Route Service 3D (ERS3D). Fig. 4 shows a sequence diagram for this service. The only difference between the ERS3D and the ERS is that the request, with the AvoidLists, is not forwarded to a normal RS, but to an RS3D. This way the RouteGeometry is 3D instead of 2D. Apart from this, the process runs the same way as described above.

Thus, both features that have been proposed here and that go far beyond the usual routing service, are available as one single service that harmonizes with other SDI components very well and can be deployed in a specific application scenario.

The only desirable extension could be to have the ERS3D generate 3D geometries for the visualization of the AvoidAreas directly. That however, would not foolow any standards so we forwent this in order to stay conform to the OpenLS specification. Therefore the Java Client has to access the WFS by itself. Then he can visualize the AvoidAreas correspondingly on its own. The needed elevation points are at the moment calculated by simple 3D picking in the scenegraph and the polygons are extruded upwards by several meters. (Fig. 5).

6 Outlook

The concept of the ERS3D uses individual standardized services as building blocks for more complex services and functionalities. This is done by aggregation resp. by chaining services. This demonstrates the usefulness of standardized modular building blocks and proves that the OGC has a lot more to offer than just web-mapping. In the future more and more emphasis will be laid on the 3D visualization of route planning and navigation support. Several manufacturers of car navigation systems are already working in this direction. Special aspects such as mobile 3D route navigation using landmarks etc. are being examined in the frame of the new BMBF project "MONA3D - Mobile Navigation 3D" .

Furhter the performance of the RS3D will have to be tested, especially when long routes have to be matched to the DEM. For this, a large amount of data has to be accessed from a WFS, the DEM has to be created, and height information has to be calculated to the route.

Additionally an implementation of the rest of the OpenLS services is currently in progress and the OpenLS Location Utility Service (Geocoder/ReverseGeocoder) can be used within the ERS3D and the ERS for geocoding start- and destination points. Further 3D related extensions of all OpenLS Core Services (not only RS) would be desirable and are being discussed in the OGC. Preliminary work has already been conducted by Zlatanova and Verbree (2005). But this work is more conceptual and without detailed suggestions for extending interfaces resp. for implementation. In addition, an evaluation should be done to assess at what granularity it is wise to have calculations done internally by a Web Processing Service (WPS), for example when it comes to height information derivation of the route geometry. Further extension possibilities could be the support of indoor navigation and optimizing the 3D visualization for that on mobile displays through the proper use of landmarks (Mohan and Zipf 2007).

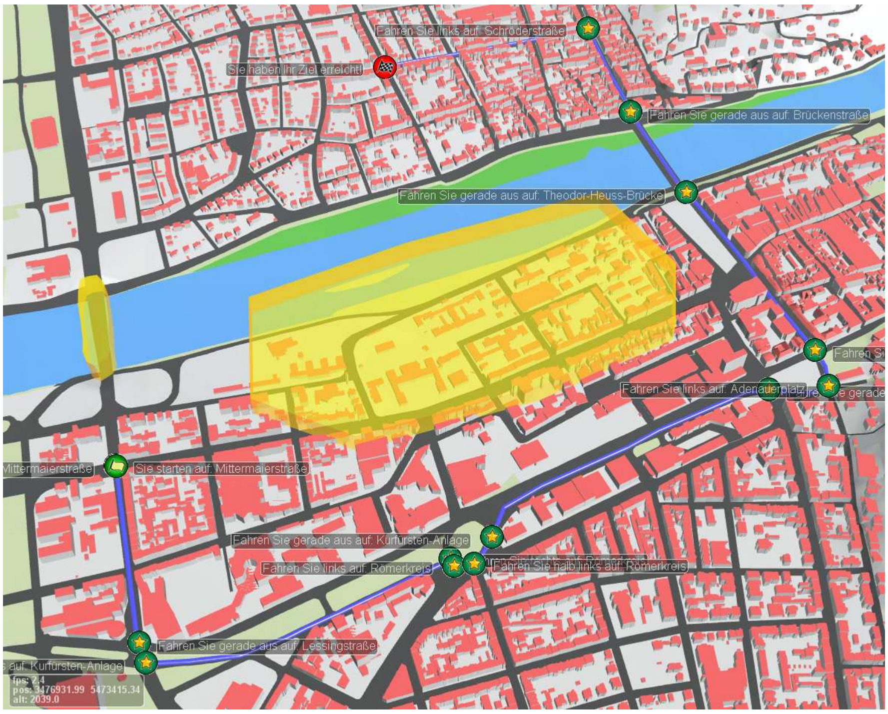

Fig. 5: Graphical output of the ERS3D in the 3DViewer. The danger zones are being displayed as half transparent extruded polygons.

Acknowledgements

We would like to thank our colleagues for their help and inspiration. We also thank the Land Surveying Office (Vermessungsamt) Heidelberg and the European Media Laboratory for supplying us with base data. This work has been funded by the Klaus-Tschira-Stiftung (KTS) gGmbH and by the BMBF-Project „OK-GIS".

References

Biermann J, Gervens T, Henke S (2005): Analyse der Nutzungszenarien und Aktivitäten der Feuerwehr. Internes Projektdokument. Projekt OK-GIS.

Bock N (2007 in progrss): Entwicklung eines W3DS Client auf Basis von MS .Net. Diploma Thesis. Fachhochschule Mainz.

Dijkstra EW (1959): Note On Two Problems In Connexion With Graphs, In: Numerische Mathematik, Springer Verlag pp. 269-271, vol. 1, Berlin.

Fischer M, Basanow J, Zipf A (2006): Mainz Mobile 3D - A PDA based client for the OGC Web 3D Service and corresponding server. International Workshop on 3D Geoinformation 2006 (3DGeoInfo'06). Kuala Lumpur. Malaysia.

Fitzke J, Greve K, Müller M, Poth A (2004): Building SDIs with Free Software - the deegree Project. In: Proceedings of GSDI- 7, Bangalore, India.

Hansen S, Richter KF, Klippel A (2006): Landmarks in OpenLS - A Data Structure for Cognitive Ergonomic Route Directions. In M. Raubal, H. Miller A, Frank U, Goodchild MF (Eds.) Geographic Information Science - Fourth International Conference, GIScience 2006, pp. 128–144. Springer, Berlin.

Kessler C, Schade S, Starke A, Tegtmeyer S, Walkowski A (2003): ariadne – GI-Dienste für Notfall-Management-Systeme. In: BERNARD, L., A. SLIWINSKI & K. SENKLER : Geodaten- und Geodienste-Infrastrukturen – von der Forschung zur praktischen Anwendung. Münster: 297-309. (= IfGIprints, Bd. 18)

Kray C, Elting C, Laakso K, Coors V (2003): Presenting Route Instructions on Mobile Devices. Proceedings of IUI'03, ACM Press, New York, NY, pp. 117-124.

Mohan SK, Zipf A. (2007 accepted): Improving the support of indoor evacuation through landmarks on mobile displays. GI4DM 2007. Joint CIG/ISPRS Conference on Geomatics for Disaster and Risk Management. Toronto, Canada.

Neis P (2006): Routenplaner für einen Emergency Route Service auf Basis der OpenLS Spezifikation. Diploma Thesis, FH Mainz.

Neis P (2007 in progress): Die OpenLS Core Services. In Andrae C, Fitzke J, Zipf A (Eds.) OpenGIS Essentials - Spezifikationen des OGC im Überblick, Wichmann Hüthig Verlag. Heidelberg.

Ringlstetter C. (2003): OCR-Korrektur und Bestimmung von Levenshtein-Gewichten. Magisterarbeit. LMU, München.

Schilling, A., Basanow, J., Zipf, A. (2007 accepted): VECTOR BASED MAPPING OF POLYGONS ON IRREGULAR TERRAIN MESHES FOR WEB 3D MAP SERVICES. 3rd International Conference on Web Information Systems and Technologies (WEBIST). Barcelona, Spain. March 2007.

Schilling, A., Zipf, A. (2002): Generation of VRML City Models for Focus Based Tour Animations - Integration, Modeling and Presentation of Heterogeneous Geo-Data Sources. In Web3D Conference, 9-12.03.2003, Saint Malo, France.

Weiser A., Neis P. and Zipf A. (2006): Orchestrierung von OGC Web Diensten im Katastrophenmanagement - am Beispiel eines Emergency Route Service auf Basis der OpenLS Spezifikation. In: GIS - Zeitschrift für Geoinformatik. Abc-Verlag. 09/2006. pp. 35-41.

Weiser A., Zipf A. (2007 accepted): Web Service Orchestration (WSO) of OGC Web Services (OWS) for Disaster Management. GI4DM 2007. Joint CIG/ISPRS Conference on Geomatics for Disaster and Risk Management. Toronto, Canada.

Zlatanova S, Verbree E (2005) The third dimension in LBS: the steps to go. In: Geowissenschaftliche Mitteilungen, Heft Nr. 74, 2005, pp. 185-190.

Zipf A (2004): Mobile Anwendungen auf Basis von Geodateninfrastrukturen - von LBS zu UbiGIS. In: Bernard, L., Fitzke J, Wagner R (eds): Geodateninfrastrukturen. Wichmann Verlag. Heidelberg.